QoS networking with Linux

Werner Almesberger
EPFL ICA

February 17, 1999

`ftp://lrcftp.epfl.ch/pub/people/almesber/slc99/`

Overview

- What is Quality of Service (QoS) ?
- QoS architectures
  - The telephony way: ATM
  - Likewise, but for IP: RSVP
  - Second thoughts about scalability: Differentiated Services
- QoS support on Linux
  - ATM on Linux
  - Linux traffic control
  - RSVP and Differentiated Services on Linux
- QoS-related research on Linux:
  - RSVP over ATM
  - The Scalable Reservation Protocol (SRP)
- Conclusion and references

What is Quality of Service ?

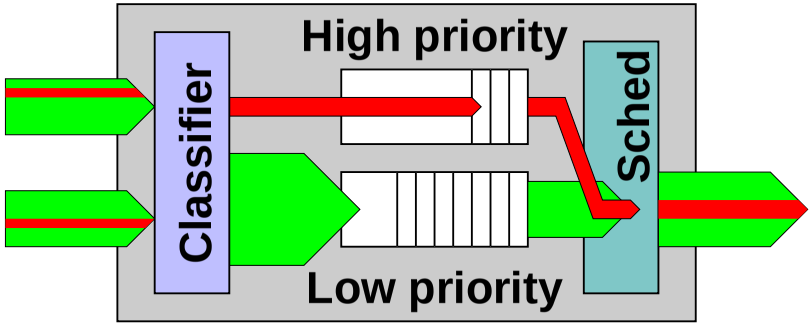
- Not all applications have the same needs, e.g.
  - Telephony wants low delay and dependable bandwidth
  - FTP wants throughput
  - E-Mail is happy with whatever is available

- QoS: provide the network service the application needs

- Assumption: it is not possible to build large networks with low delay, high throughput, few losses, etc., at an acceptable price

How is it done ?

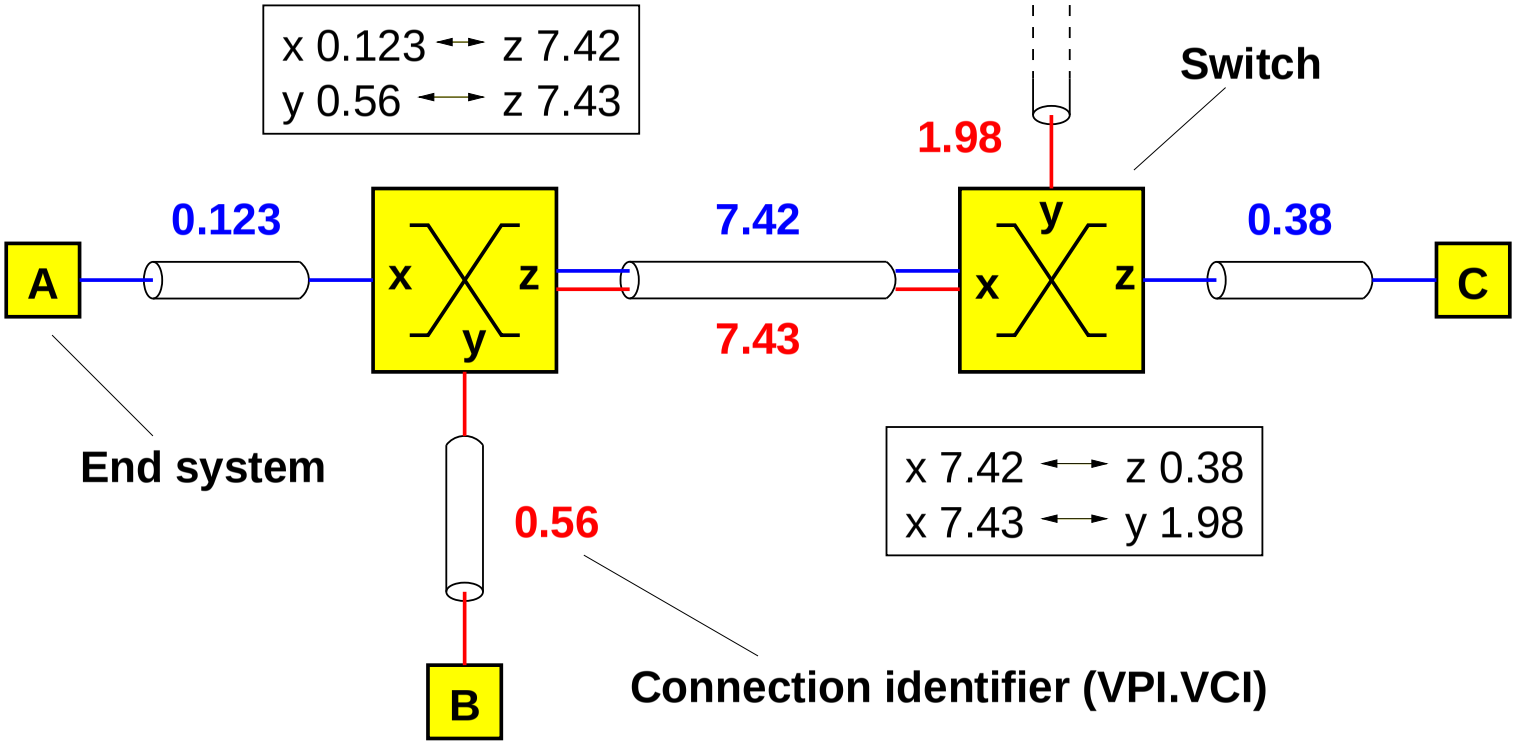
- Service differentiation
  - Classify traffic based on required service
  - Apply different processing to classes



- Dependable service
  - Provisioning
  - Isolation from other traffic (e.g. telephony)
  - Relative fairness (e.g. TCP)

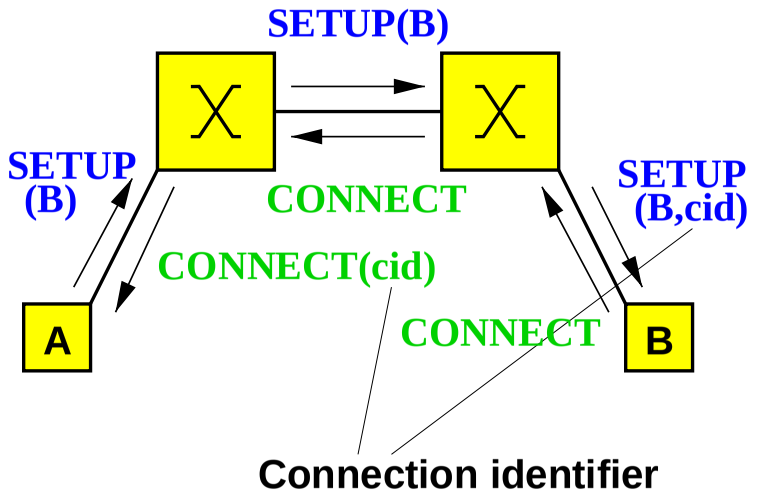
Asynchronous Transfer Mode

- Connection-oriented network technology for integrated services
- Evolved from narrow-band ISDN
- Supported by telecom industry



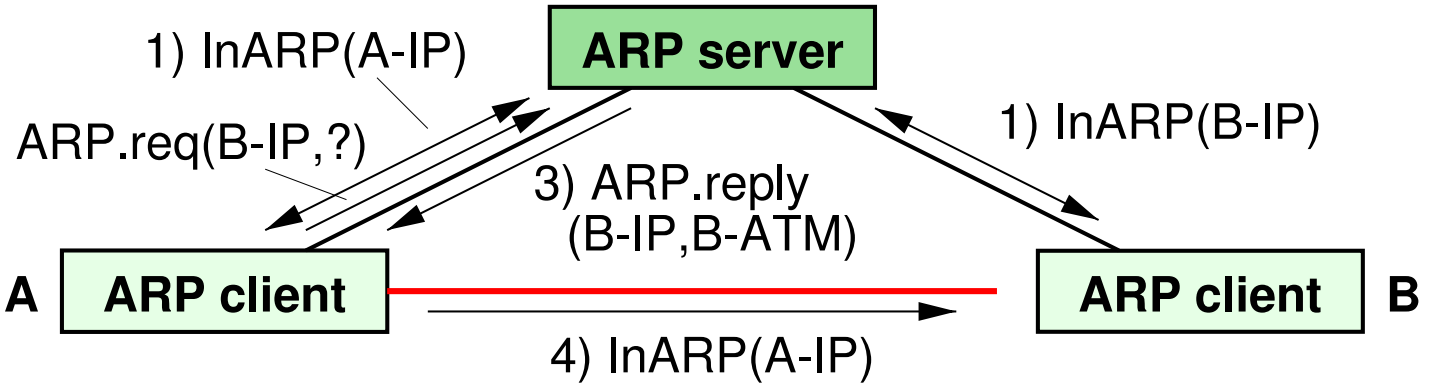
QoS with ATM

- Signaling for "automatic" connection setup
- Sophisticated QoS architecture
  - Explicit reservation of bandwidth
  - Traffic classes:
    - Best effort
    - Constant Bit Rate (CBR; peak)
    - Variable Bit Rate (VBR; peak, average, burst size)
    - Available Bit Rate (ABR; with congestion control; UNI 4.0)
  - Delay control (UNI 4.0)



ATM and IP

- Most applications are on IP
- ATM is independent from IP
  - IP packets encapsulated in ATM "packets"
  - IP to ATM address resolution
    - Classical IP over ATM
    - LAN Emulation
    - NHRP/MPOA

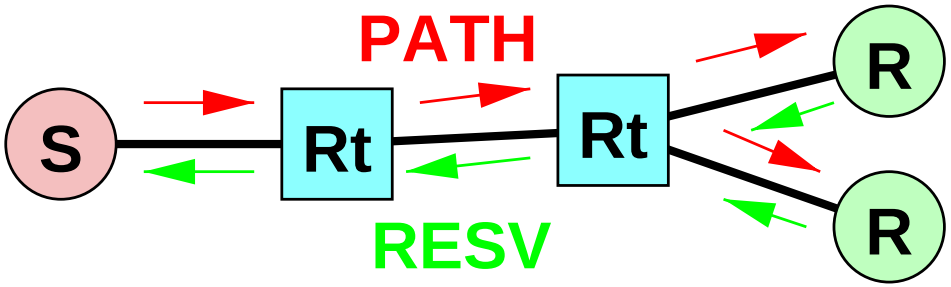


Resource ReSerVation Prot.

- General design similar to ATM:
  - Connection-oriented (flows)
  - Reservations are made for individual flows
  - Slight difference: "soft state"
- Designed by IETF for IP ($\rightarrow$ fewer interoperability issues)
- Traffic classes
  - Guaranteed service: bounded delay at given rate
  - Controlled load: behaves like unloaded network
- Functional blocks in an RSVP node:
  - Classifier to select flows
  - Policing (optional)
  - Packet scheduler

RSVP (continued)

- Suitability for multicast major design goal
  - Reservations initiated by receiver
  - Reservations are merged on the way to the sender
  - Allows for heterogenous reservations



Flashback: Priorities

History

- RFC791 (September '81) defines "Precedence"
- Straightforward and efficient concept
- Implemented in most routers
- Typical use: network control traffic (e.g. routing)

Problems

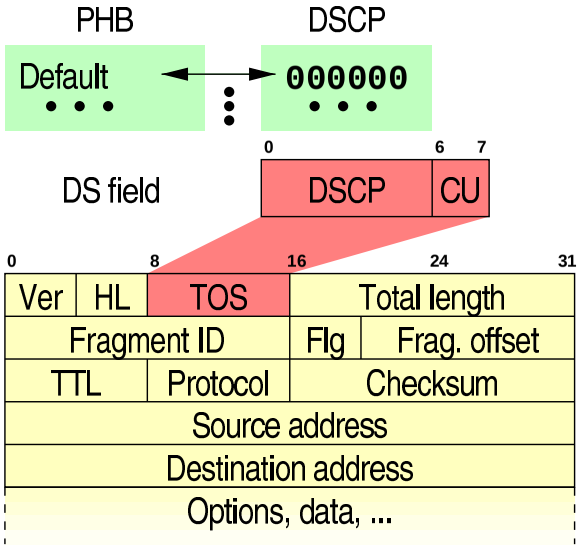
- Meaning not clear (drop, delay, ... ?)
- One-dimensional
- End-to-end service definition difficult

Differentiated Services

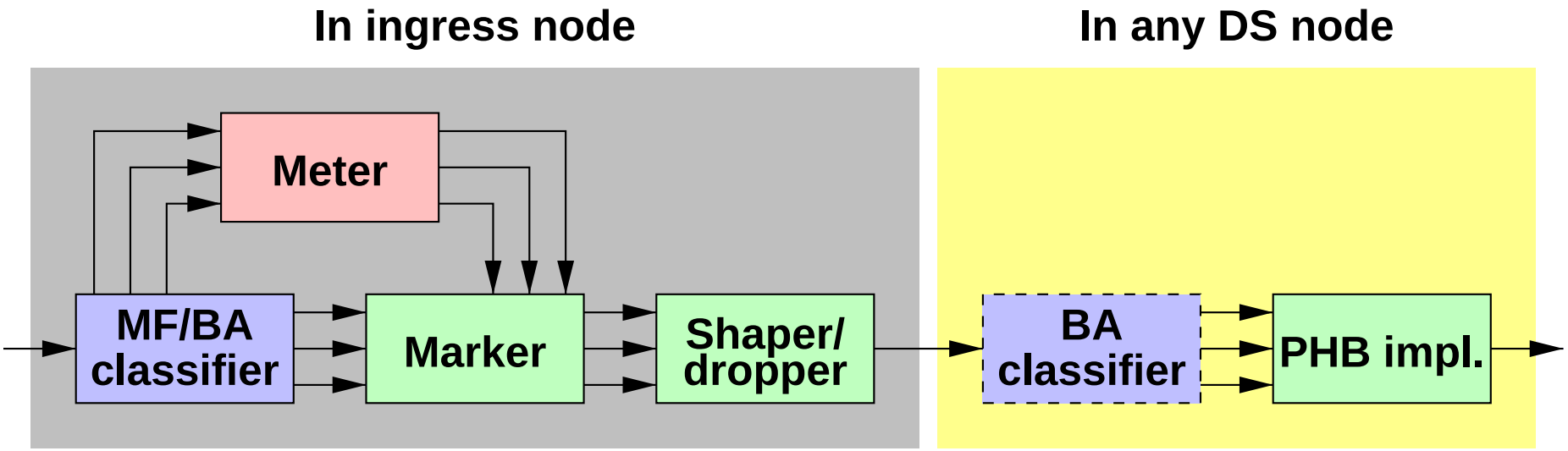
- Motivation and history:
  - RSVP does not scale well for large numbers of flows
  - RSVP expensive to implement
  - Market demands service differentiation (e.g. for VPNs)
  - Router vendors are starting to deploy priority-based proprietary solutions
  - IETF diffserv WG completed RFC with basic design in one year !
- Generalized precedence concept
  - Each packet selects a specific per-hop behaviour (PHB)
  - Up to 64 different PHBs can be supported on a link
  - Only externally observable forwarding behaviour is standardized

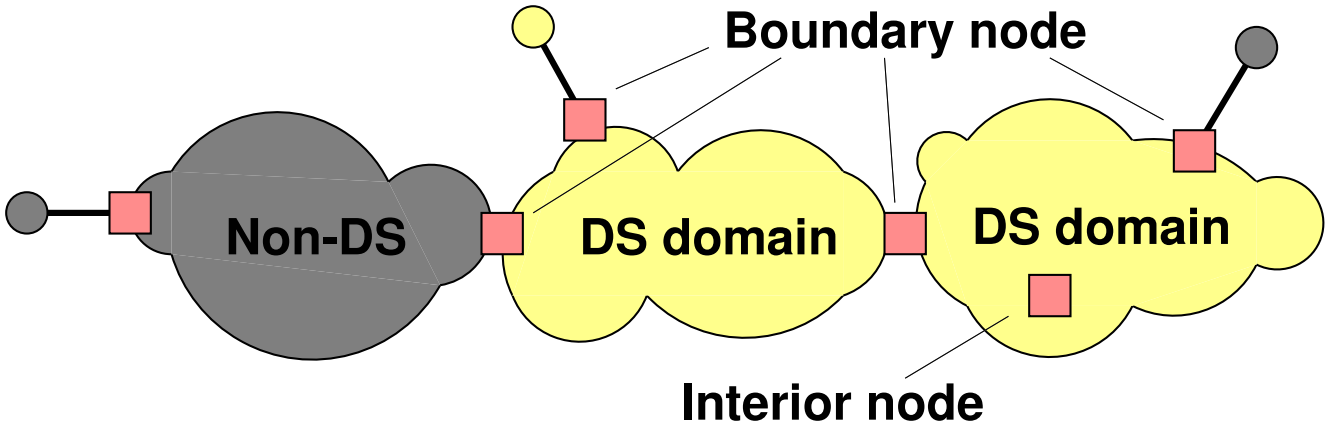
Diffserv (continued)

- Modular PHB concept
  - Expedited Forwarding (EF): single high priority, e.g. for "Virtual Leased Line"
  - Assured Forwarding (AF): matrix of delay and drop priorities
- Inexpensive to implement
  - Packets are distinguished only by the DS field in the IP header
  - In IPv4, the DS field occupies the space previously used for the TOS (Type Of Service) byte
  - All packets with the same DS field are treated as a single aggregate



Diffserv nodes

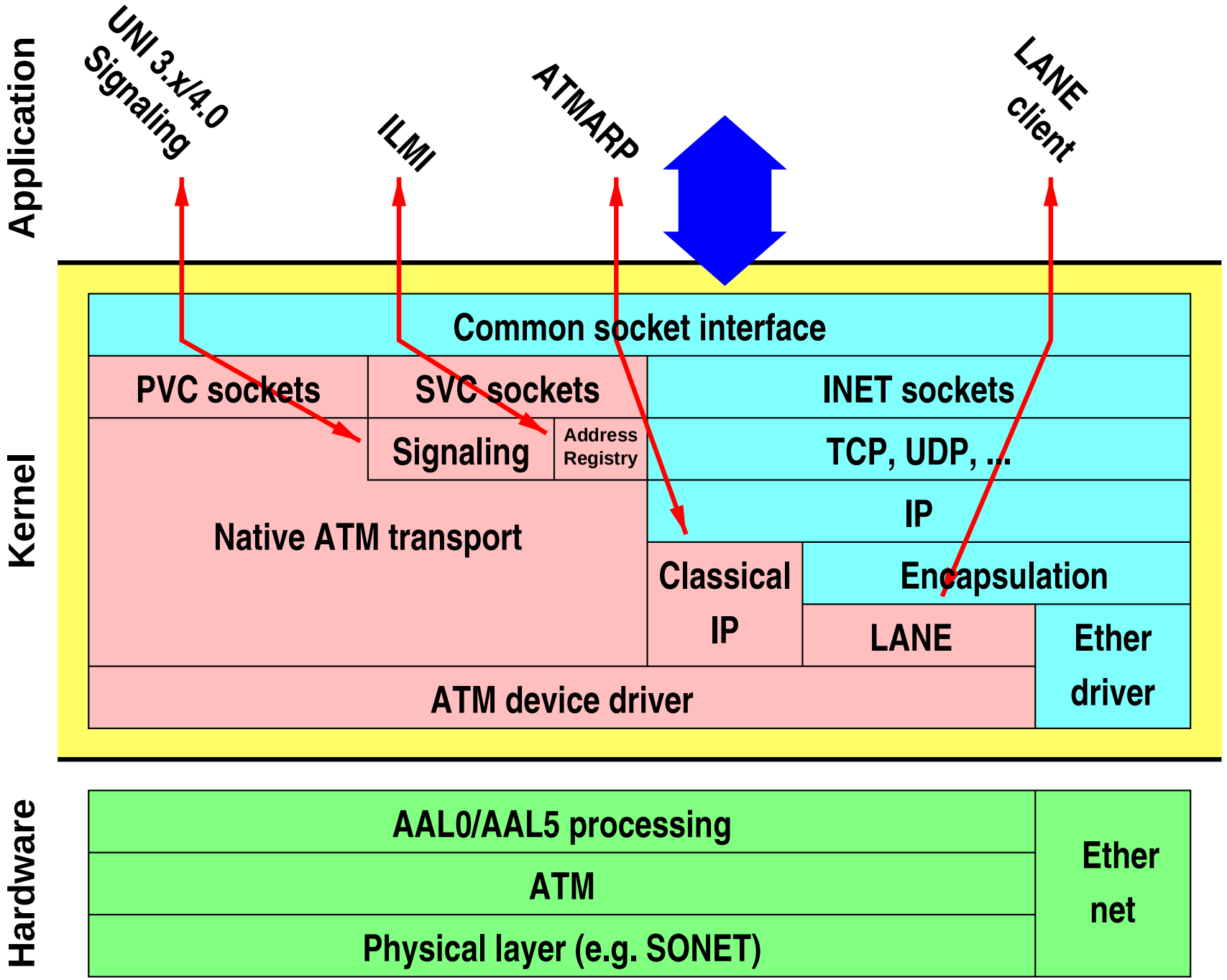


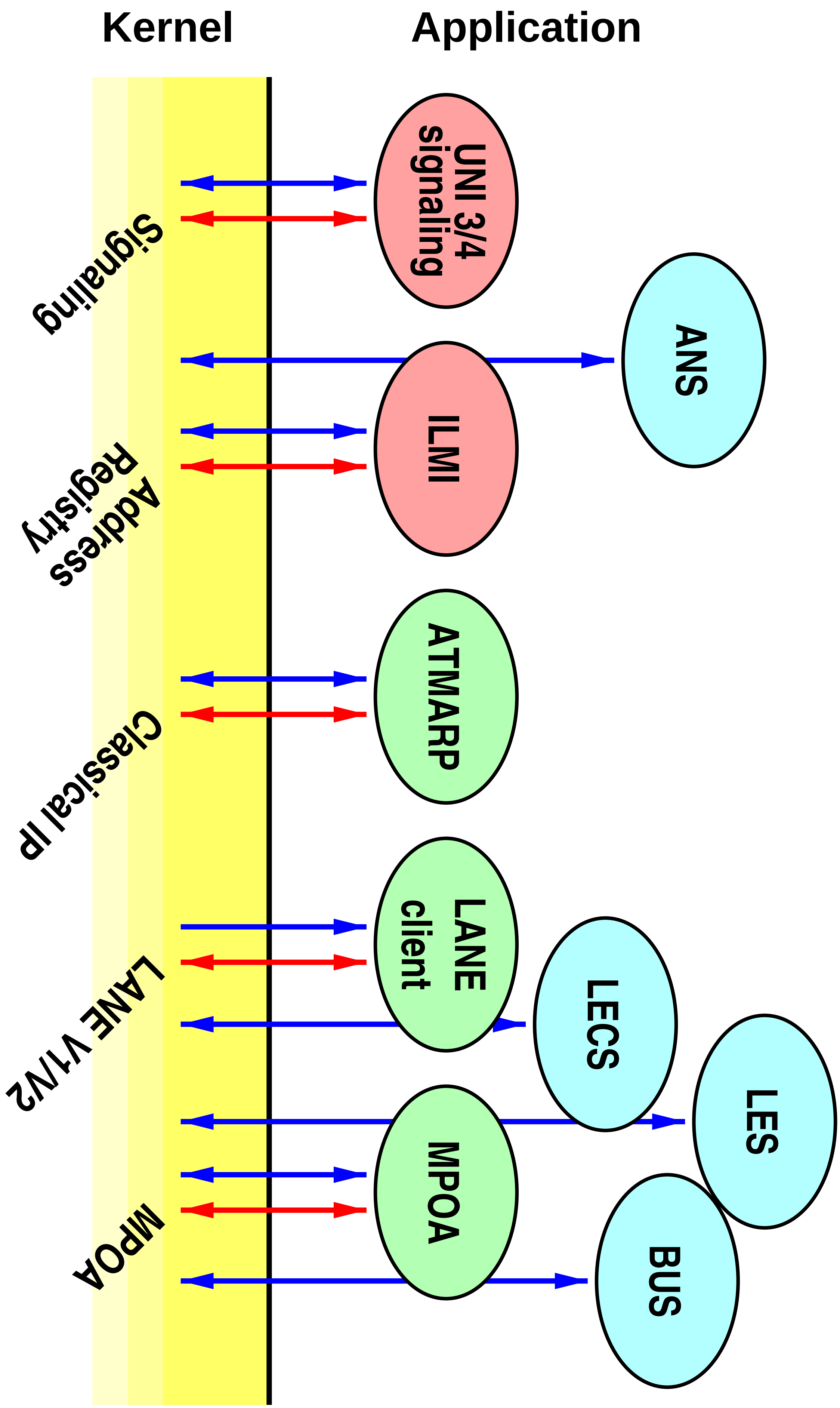


ATM on Linux

- History
  - Goal: State of the art implementation of ATM protocols
    - Platform for research
    - Reference material for education
    - Visibility
  - Project started 1994/1995 at EPFL
  - Source code fully available
  - Turned into global effort

- Status
  - UNI 3.0, 3.1, and 4.0 unicast signaling
  - Classical IP over ATM (RFC1577), LAN Emulation V1/V2, MPOA
  - Support for UBR and CBR (Unspecified/Constant Bit Rate)

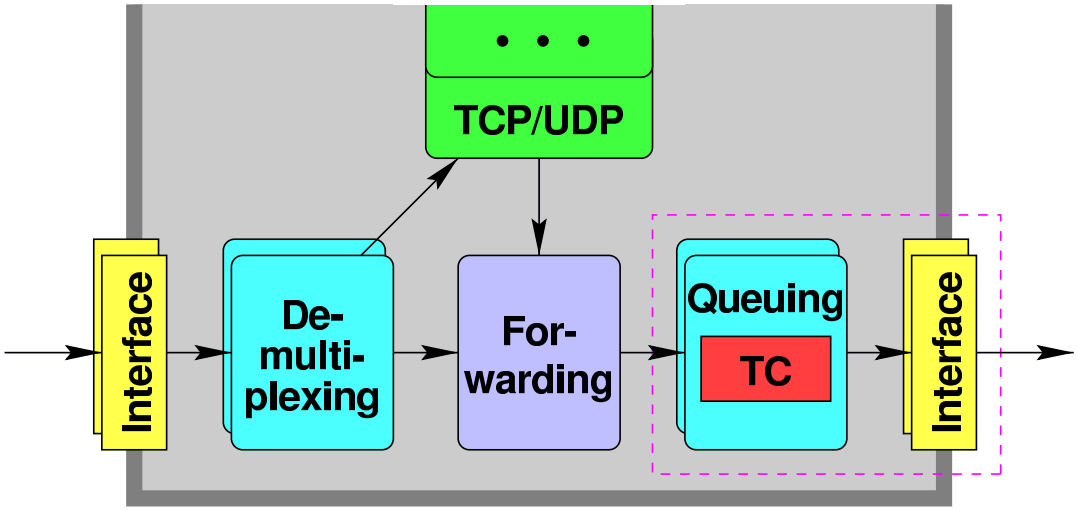

Application
UNI 3.x/4.0 Signaling
ILMI
ATMARP
LANE client
Kernel
Common socket interface
PVC sockets
SVC sockets
INET sockets
Signaling
Address Registry
TCP, UDP, ...
IP
Native ATM transport
Classical IP
Encapsulation
LANE
Ether driver
ATM device driver
Hardware
AAL0/AAL5 processing
ATM
Physical layer (e.g. SONET)
Ether net



Kernel
Application
Signaling
Address Registry
Classical IP
LANE V1/V2
MPOA
UNI 3/4 signaling
ANS
ILMI
ATMARP
LANE client
LECS
LES
MPOA
BUS

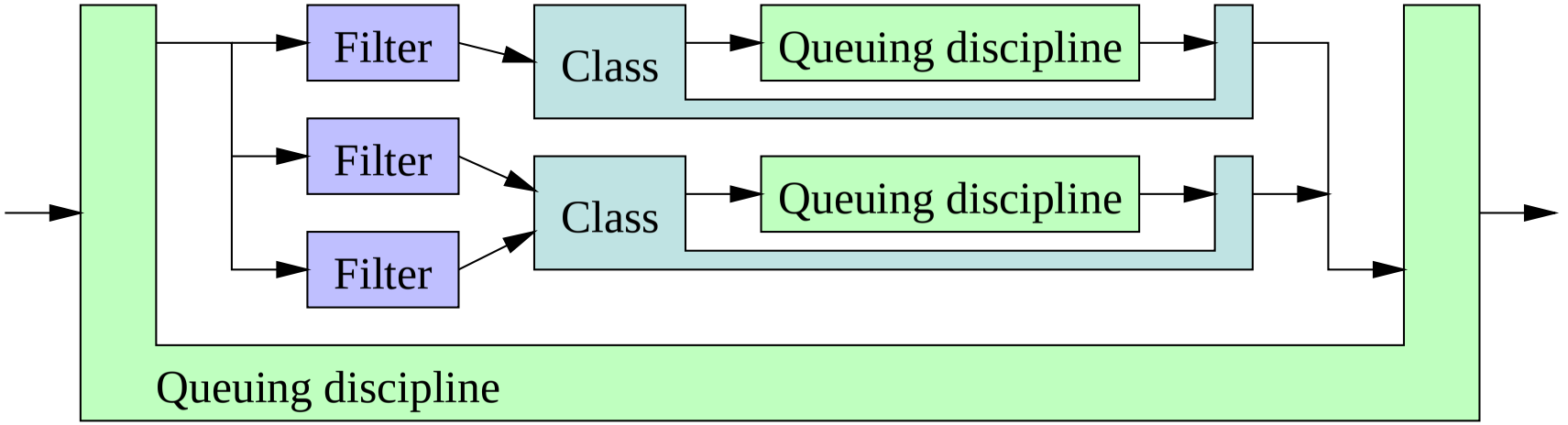
Linux traffic control

- Added in 2.1 kernels (by Alexey Kuznetsov)
- Modular framework for building (almost) arbitrary traffic control functions
- Classification, scheduling, policing

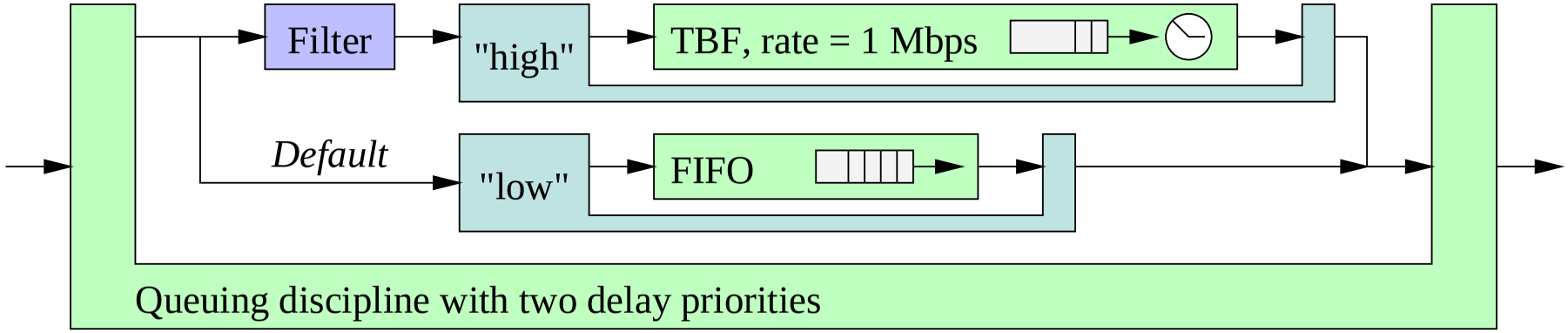


Traffic control elements

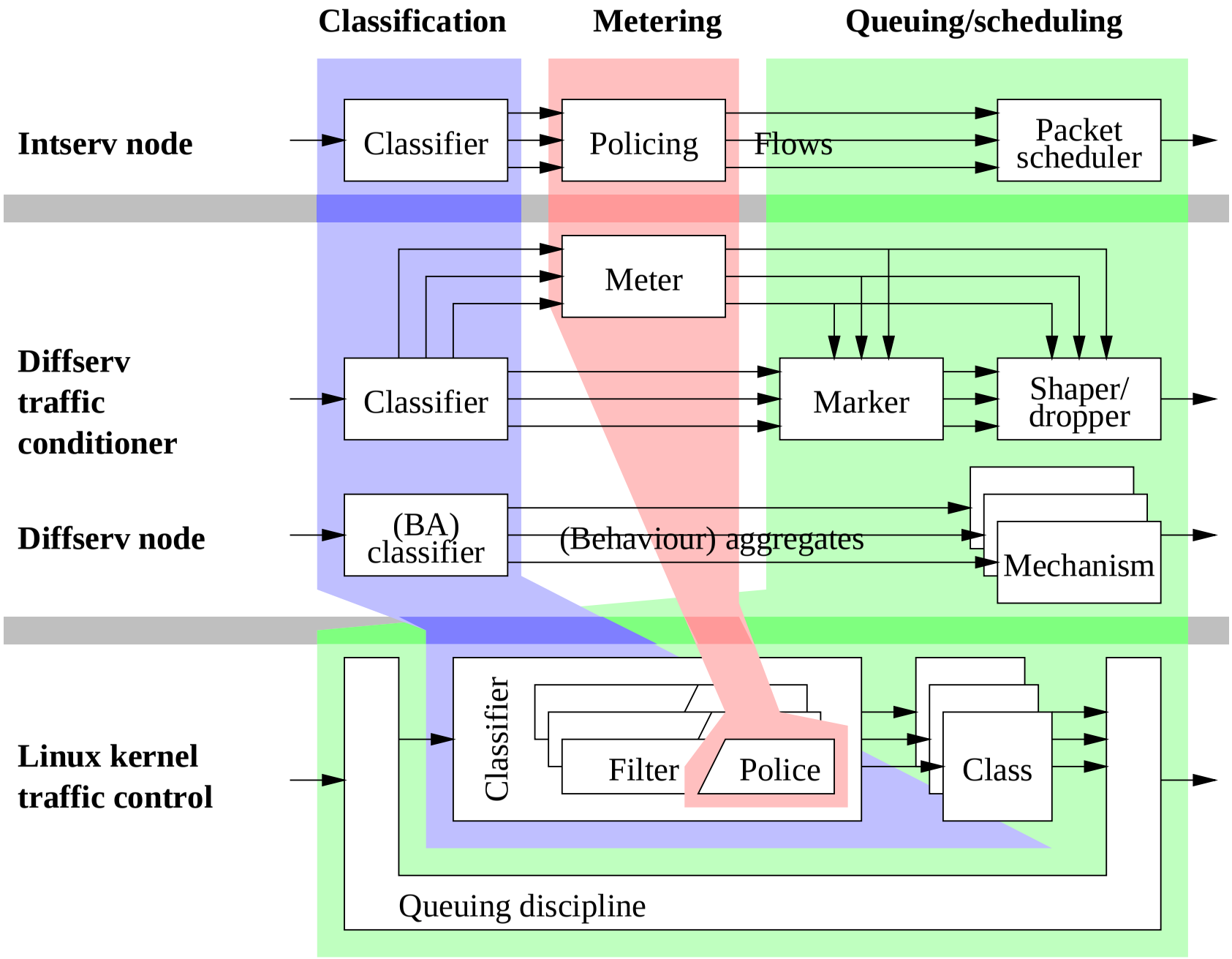
- Queuing disciplines define general semantics
  (e.g. FIFO, PRIO, TBF, CBQ, ...)
- Different classes implement different behaviour
- Packets are attributed to classes by filters
  (e.g. RSVP classifier)
- Classes may in turn contain queuing disciplines



Combination of queuing disc's

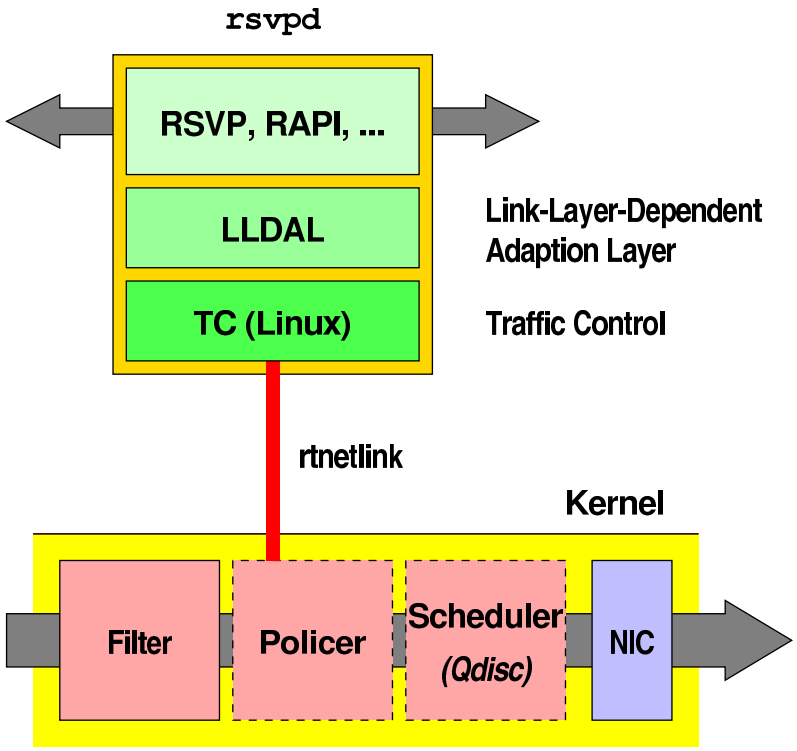


- Example:
  - High-priority traffic is always scheduled before low-priority traffic
  - TBF (Token-Bucket Filter) limits the rate of high-priority traffic so that it can't starve low-priority traffic


Classification
Metering
Queuing/scheduling
Intserv node
Classifier
Policing
Flows
Packet scheduler
Meter
Diffserv traffic conditioner
Classifier
Marker
Shaper/ dropper
Diffserv node
(BA) classifier
(Behaviour) aggregates
Mechanism
Linux kernel traffic control
Classifier
Filter
Police
Class
Queuing discipline

RSVP on Linux

- Several independent implementations/ports
- Typically based on ISI `rsvpd`
- Port by Alexey Kuznetsov very tightly integrated with kernel traffic control



Diffserv on Linux

- Time-line:
  - December '98: first prototype for 2.1.129 by Alexey Kuznetsov, Jamal Hadi Salim, and Werner Almesberger
  - February '99: design corrections/extensions
  - Later in '99: integration into 2.2 or 2.3 kernel

- Classification/marking functionality:
  - DS-capable host
  - DS boundary node
    - non-DS→DS
    - Limited: DS→DS
  - DS interior node

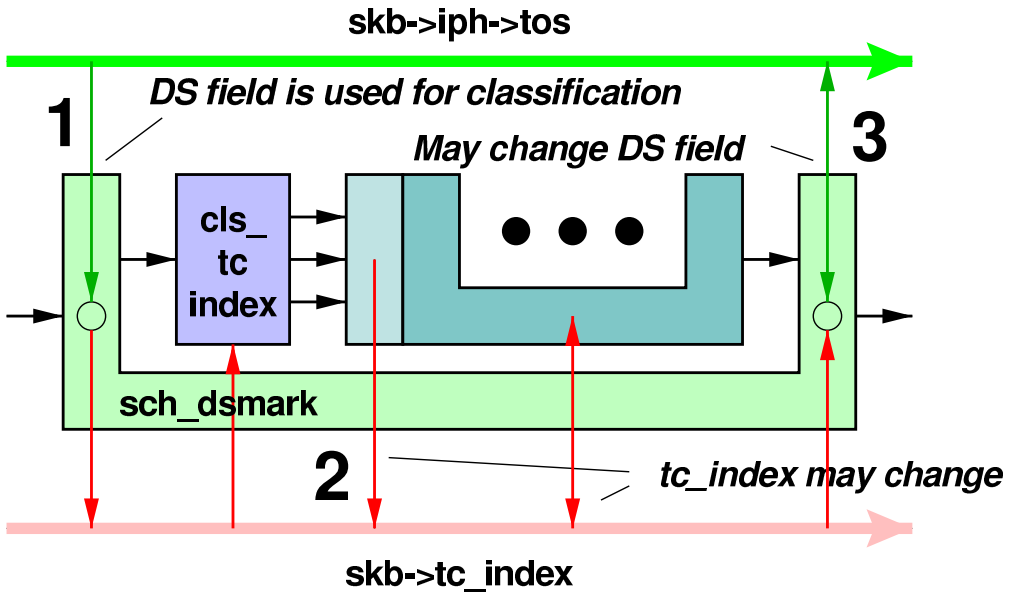
Diffserv on Linux (cont'd)

- Defining per-hop behaviours
  - Preserved modular concept of traffic control
  - Example scripts for Expedited Forwarding (EF) and Assured Forwarding (AF)

- Defining classifiers
  - Preserved modular concept of traffic control
  - Example scripts for Behaviour Aggregate Classifier (BAC) and various edge configurations

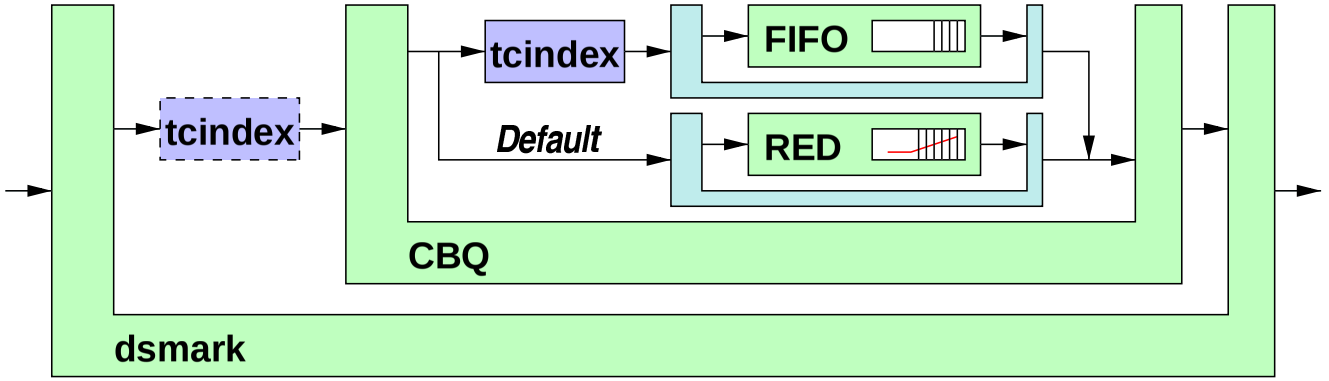
Diffserv framework

- New socket buffer field `skb->tc_index` to store classification result
- New classifier `tcindex` to use `skb->tc_index` in classification
- New queuing discipline `dsmark`
  - Copies the DS field into `skb->tc_index` (**1**)
  - Stores the classification result in `skb->tc_index` (**2**)
  - Updates the DS field based on `skb->tc_index` (**3**)



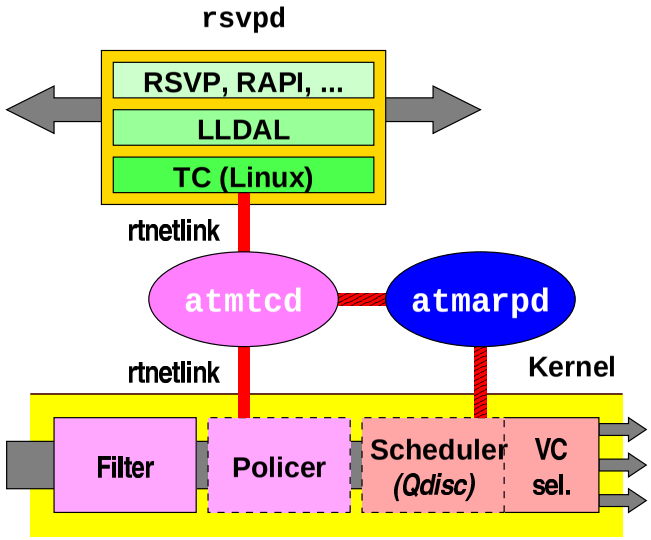
Implementing a PHB

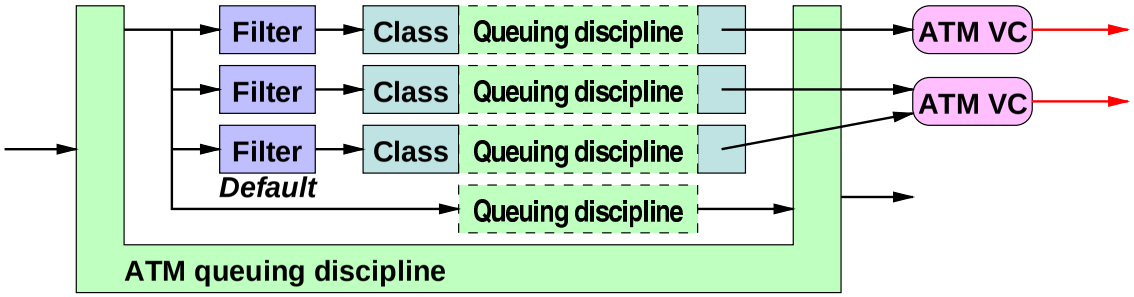
- New queuing discipline `GRED`
  - Generalized RED (Random Early Detection)
  - One queue with multiple drop priorities
  - Necessary to implement Assured Forwarding (3 drop priorities)
- Example: Expedited Forwarding
  - CBQ maintains priority and controls rates
  - RED ensures fairness for best-effort



Research: RSVP over ATM

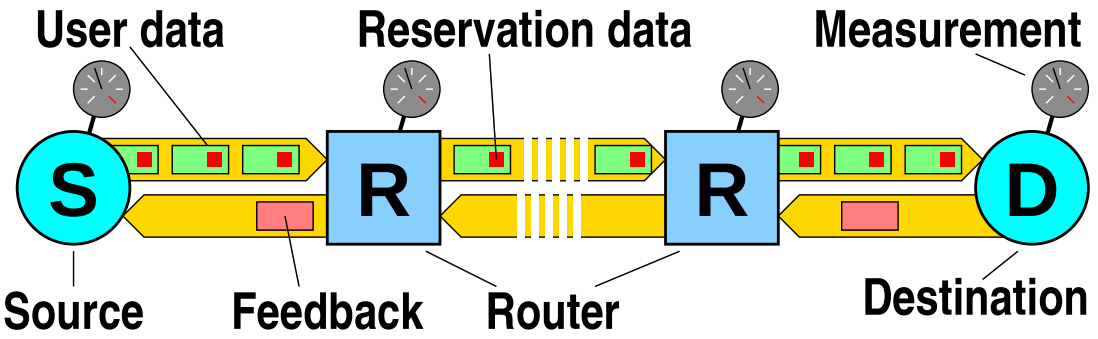
- `atmtcd`
  - Translates INTSERV traffic parameters to ATM
  - Resolves addresses via `atmarpd`
  - Manages VCs
- ATM queuing discipline
  - Directs classes to ATM VCs
  - Can be used like any other queuing discipline





Research: SRP

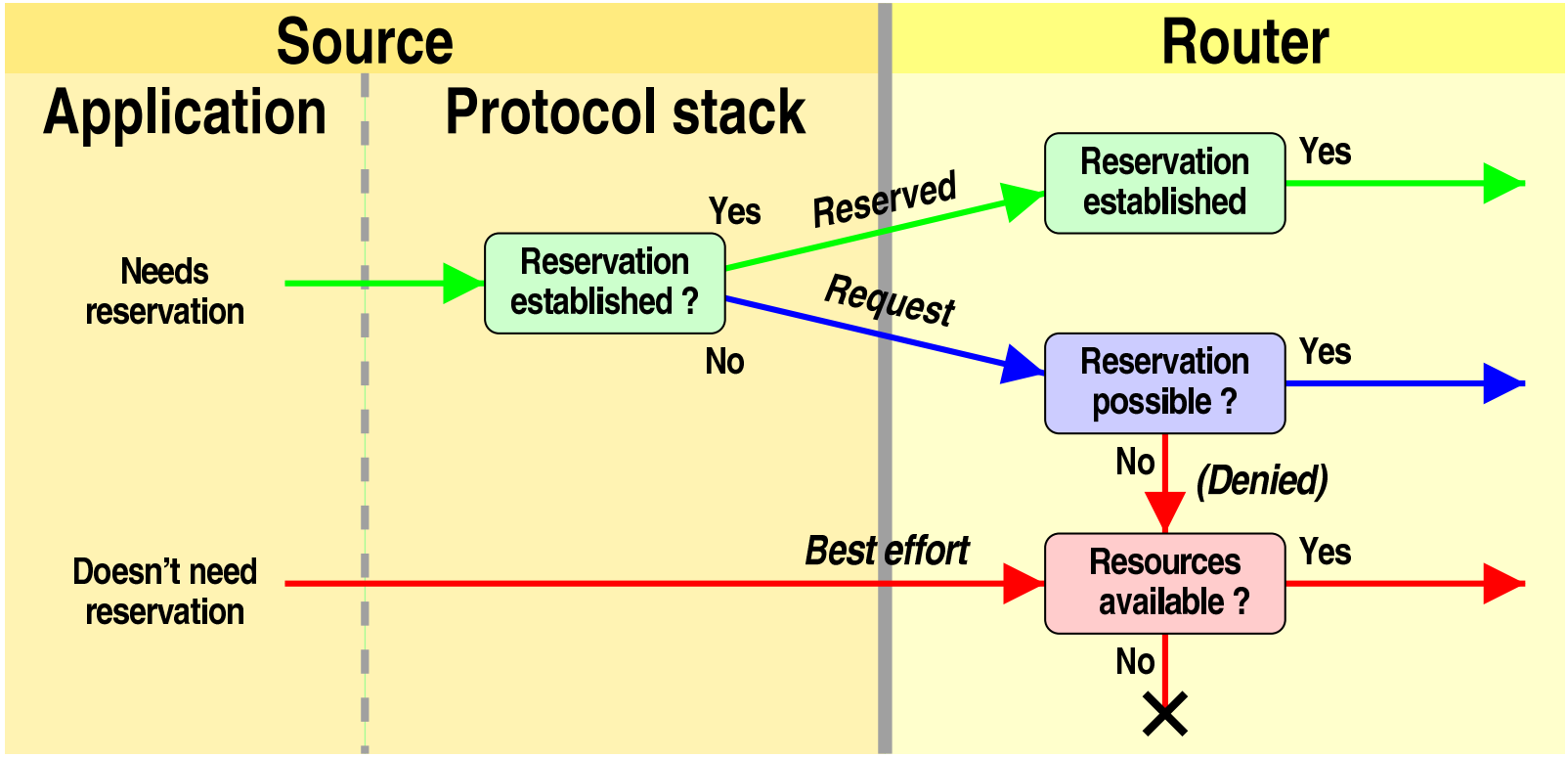
- Scalable Reservation Protocol
- Diffserv scales well but depends on external elements to control resource use
- SRP uses a single protocol end-to-end and scales well for large numbers of flows



- Reservation information (2 bits) is sent in forward direction with data
- Destination occasionally sends feedback to source
- All systems monitor the aggregate traffic

Research: SRP (continued)

Three packet types: Reserved, Request, and Best effort. Uses Diffserv for Reserved and Request.



Conclusion

- There are several competing QoS architectures
- Evolution: Telephony $\rightarrow$ IP $\rightarrow$ Aggregation $\rightarrow$ ...
- Linux supports all major QoS architectures
- Linux traffic control can be easily extended

References

- ATM on Linux (code, documentation, mailing list)
  `http://lrcwww.epfl.ch/linux-atm/`
- RSVP on Linux (code; mirror)
  `ftp://ftp.funet.fi/mirrors/ftp.inr.ac.ru/ip-routing/rsvp/`
- RSVP over ATM (project page)
  `http://www.telscom.ch/diana/`
- Linux traffic control (paper)
  `ftp://lrcftp.epfl.ch/pub/people/almesber/pub/tcio-current.ps.gz`
- Diffserv on Linux (code, mailing list)
  `http://lrcwww.epfl.ch/linux-diffserv/`
- SRP (papers and simulation)
  `http://lrcwww.epfl.ch/srp/`